Tropical Fish Emporium

Group Project: Final Submission Portfolio

Andrew Clos & Haley Woehrle
Oregon State University
CS 340: Section 401

Table of Contents

Executive Summary

Design Changes

Early on we had planned on having a 'stores' entity, but it was suggested to us that we should remove this because of the added complication it could present. Additionally, we had many (6-10) attributes for each entity, and more than the required number of foreign keys. We cut these down as well to simplify the structure of our database. Furthermore, we had two, many to many relationships. Over the course of several weeks, we simplified our database significantly, so that all requirements were met and so that no extra effort would be spent on extraneous relationships and features.

Based on peer feedback, we simplified our implementation of our UPDATE and DELETE. We had originally planned on allowing a user to perform an UPDATE or DELETE operation on all four tables in the database, but this would have required a lot of repetitive coding and would have potentially been more confusing to the user if we were to properly enforce foreign key constraints. (2.3 Peer Feedback, 3.3 Peer Feedback, 5.2 Peer Feedback, 6.2 Peer Feedback)

Facts About the Database

We removed productId as an attribute of Sales, Stores as an entity, and the Customer to Employee relationship. We added mandatory participation of Product to Sales relationship and of Customer to Sales. We also added a transactionDate attribute to the Sale entity so that the client can query the database to look at sales on a particular day. Additionally, our online database does not enforce the unique key combination of salesId and productID for a sales_product record. In the real world, we would enforce this unique constraint, but for evaluation purposes, we did not want the database to cause an error if a repeat sales_product entry was created. (2.1 Changes Made, 3.1 Changes Made, 5.1 Changes Made, 6.1 Changes Made)

Creation of a new sales_products entry is done manually from the manager tab. It would be preferred to have this occur automatically when a new sale is entered, but this would have required the user to enter the number of unique products that were part of that sales. This is problematic, because our database's sale entity does not have an attribute for "number of products." At this stage of development, we did not want to make a change to the fundamental structure of the sales table.

ProductIDs were found to not be auto incrementing during Step 5 review, so our DDL file was updated and the database was fixed. It was originally intended that all primary key "ids" be of an auto incrementing type, so this issue was effectively a syntax error that caused a bug for a short time.

4

Project Overview

Tropical Fish Emporium (TFE) is a small chain of fish stores in your city and they need a database to help manage their business structure. The business owner would like a better way to track the relationships between her 40 employees and their overall sales of 55 unique products. TFE has an average of 100 sales per day. All TFE customers have a customer profile so that the store can hold information about the type of products they use and need for their home aquariums. There are a total of 1500 customer profiles in the system. The website will utilize a DB backend to store, manipulate, sort, link, and aggregate different types of data used by our client, Tropical Fish Emporium.

Employees	40
Customers (profiles)	1500
Average Sales Per Day	100
Products	55
Locations	3

Table 1. TFE primary business statistics

Database Outline

Employees

Attributes
1. employeeID: int(11), primary_key, auto_increment, not NULL
2. storeID: int(11) *store id where the employee works*
3. title: varchar(255), not NULL *i.e. Assistant Manager*
4. startTime: time, not NULL
5. stopTime: time, not NULL
6. hourlyRate: decimal(16,2), not NULL *pay-rate in dollars per hour*
7. partTime: boolean, not NULL *true if employee is a part time worker*
8. empFirstName: varchar(255), not NULL
9. empLastName: varchar(255), not NULL
10. empFullName: CONSTRAINT, UNIQUE (empFirstName, empLastName)

Relationships
1. *sales:* 1:M. An employee can have many sales, but each sale will be tied to one and only one employee.

Customers

Attributes
1. <u>customerID</u>: int, primary_key, auto_increment, unique, not NULL
2. email: varchar(255) *optional if no complete profile*
3. memberSince: date/time, not NULL
4. custFirstName: varchar
5. custLastName: varchar
6. custFullName: CONSTRAINT, UNIQUE (custFirstName, custLastName)

Relationships
1. *sales:* 1:M: a customer may have many sales, but each sale can only belong to a single customer

Sales

Attributes
1. <u>saleID</u>: int, primary_key, auto_increment, not NULL
2. eID: int(11), foreign_key, not NULL *employeeID who handled the transaction*
3. cID: int(11), foreign_key, not NULL *customerID who made the purchase*
4. transactionDate: date, not NULL *date of the sale*
5. totalPurchase: decimal(16,2), not NULL *dollar amount of sale*

Relationships
1. *employees:* M:1. An employee may have many sales, but each sale must be related to one and only one employee.
2. *customers:* M:1. A customer may have many sales, but each sale must be related to one and only one customer.
3. *sales_products:* 1:M: A sale can include many different products and a product type can be part of many individual sales.

Sales_Products

Attributes
1. <u>sID</u>: int(11), primary_key, foreign_key, not NULL
2. <u>pID</u>: int(11), primary_key, foreign_key, not NULL
3. number: int(11), not NULL *number of a certain product that is part of this order*

Relationships
1. This table facilitates the M:M relationship between orders and products. Because an order may contain many products and a product can be present in many different orders, this table is needed. Furthermore, this table

makes it possible to have multiple items of the same product present on an individual order.

Products

Attributes

1. <u>productID</u>: int, primary_key, not NULL *identifies products in the system by a unique number*
2. name: varchar(255) not NULL *the product name*
3. price: decimal(16,2) *how much the product costs*

Relationships

1. *sales_products:* 1:M, a product in the inventory can be part of many different sales, and a sale can have many different types of products associated with it.
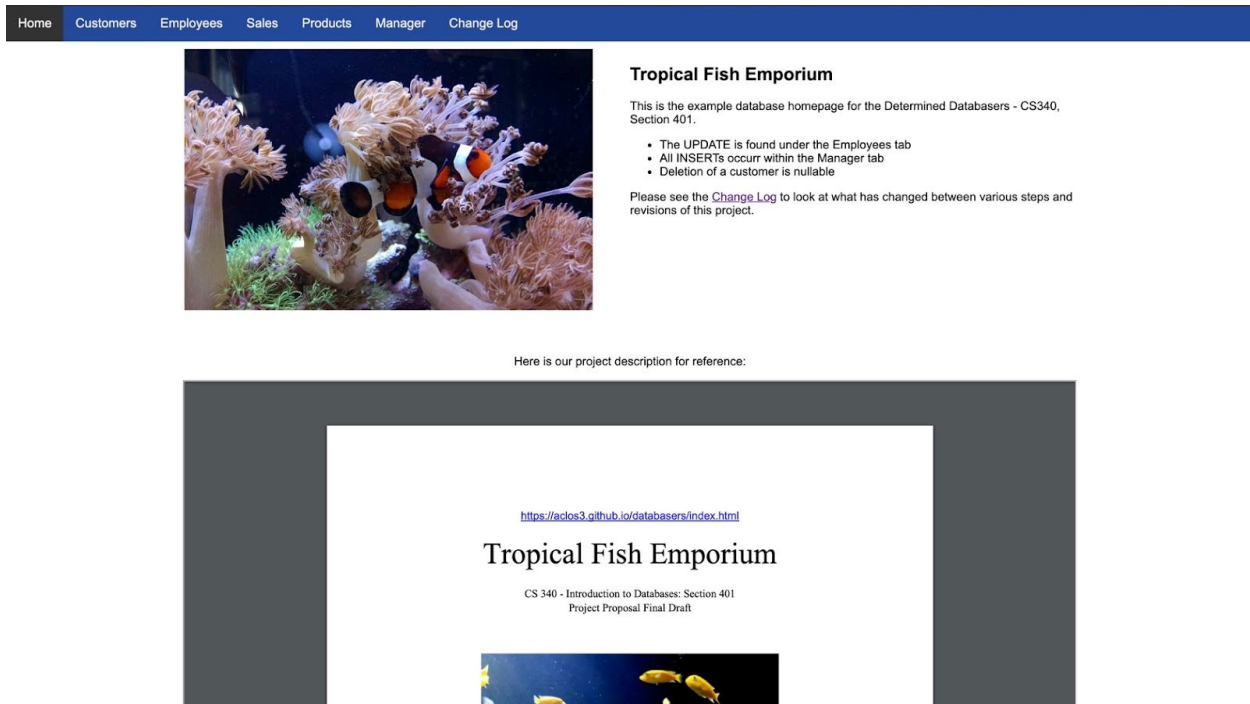
Entity-Relationship Diagram (ERD)
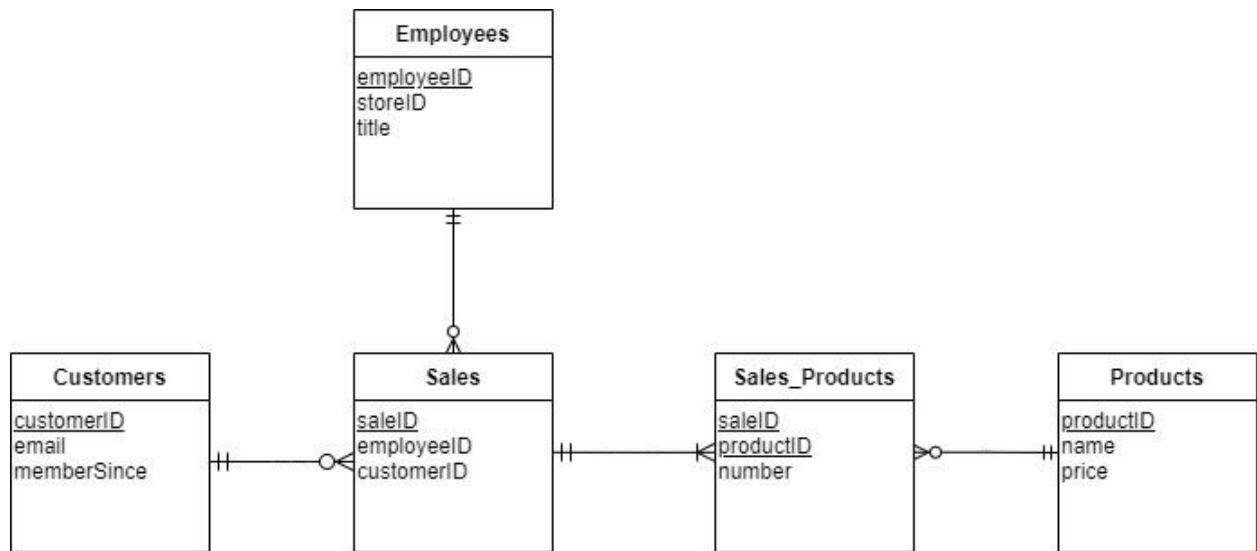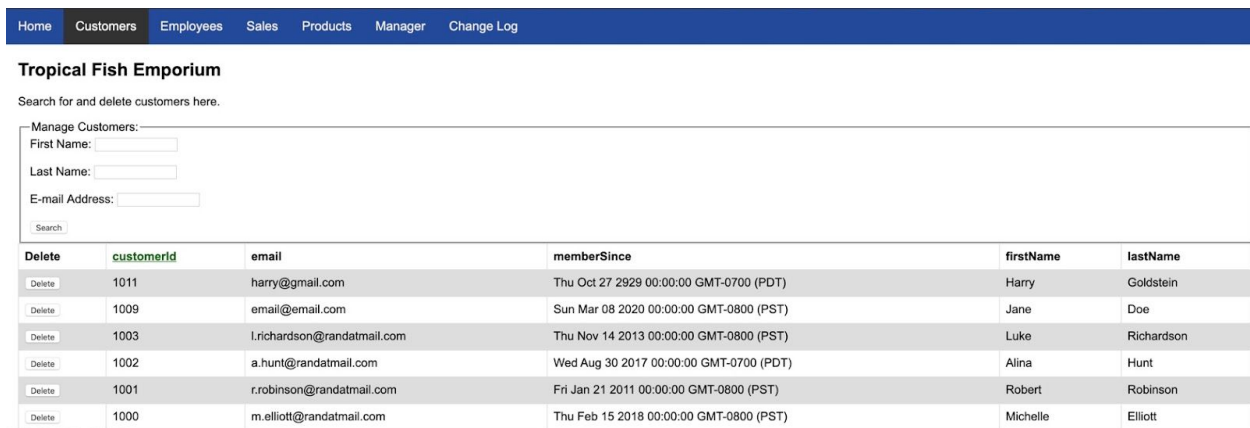
Schema

Website Screen Captures

Homepage

CRUD Step & Purpose: N/A, Display Project Portfolio

Customers Page

CRUD Step & Purpose: SEARCH/FILTER, DISPLAY/READ, & DELETE customers

Employees Page

CRUD Step & Purpose: SEARCH/FILTER, DISPLAY/READ, & EDIT/UPDATE employees

| Home | Customers | Employees | Sales | Products | Manager | Change Log |

Tropical Fish Emporium

Search for employees here.

Search Employee Here:
First Name: [ ]
Last Name: [ ]
Title: [ ]
[Search]

Update	employeeID	firstName	lastName	storeID	title	startTime	stopTime	hourlyRate	partTime
Update	2012	Tester	Employee	4342	Sorta knowledgeable fish guy	04:06:00	17:08:00	56	0
Update	2011	asd	dsa	123	asd	00:12:00	00:12:00	123	0
Update	2003	Bob	Smith	321	Fish Specialist	11:30:00	04:30:00	30	1
Update	2002	Dave	Murray	103	Fish Specialist	08:00:00	01:30:00	14.5	1
Update	2001	Silver	Craig	101	Cashier	01:00:00	05:30:00	12.5	1
Update	2000	Salane	Foster	103	Assistant Manager	09:30:00	01:00:00	20	0

© Tropical Fish Emporium

Sales Page

CRUD Step & Purpose: SEARCH/FILTER, DISPLAY/READ products & sales_products

| Home | Customers | Employees | Sales | Products | Manager | Change Log |

Tropical Fish Emporium

Search for sales here.

Search for a Sale:
Sale ID: [ ]
Employee ID (who comlpeted the sale): [ ]
Customer ID (who puchased the product): [ ]
[Search]

Sales:

saleID	employeeID	customerID	transactionDate	totalPurchase
4005	2000		Mon Feb 26 2018 00:00:00 GMT-0800 (PST)	284.98
4004	2002	1002	Tue Aug 29 2017 00:00:00 GMT-0700 (PDT)	24.99
4003	2003	1002	Thu Feb 22 2018 00:00:00 GMT-0800 (PST)	20.38
4002	2001		Tue Dec 06 2016 00:00:00 GMT-0800 (PST)	106.79
4001	2001	1000	Tue Feb 06 2018 00:00:00 GMT-0800 (PST)	83.8
4000	2002	1001	Thu Jan 11 2018 00:00:00 GMT-0800 (PST)	60.02

sales_products:

saleID	productID	quantity
4005	3005	2
4005	3004	9
4005	3003	2
4004	3003	2
4004	3002	1
4003	3002	2

Products Page

CRUD Step & Purpose: SEARCH/FILTER, DISPLAY/READ products

Manager Page

CRUD Step & Purpose: CREATE/INSERT/ADD NEW customer, CREATE/INSERT/ADD NEW employee, CREATE/INSERT/ADD NEW product, CREATE/INSERT/ADD NEW & READ/DISPLAY sale, CREATE/INSERT/ADD NEW & READ/DISPLAY sales_product

Add Product Here:
Product Name:
Price
Submit

Add Sale Here:
Employee ID (who comlpeted the sale) 2000: Salane Foster **this is a dynamically populated list of employees in the database.**

Customer ID: 1000: Michelle Elliott **this is a dynamically populated list of customers in the database.**

Transaction Date: mm/dd/yyyy

Purchase Total (including tax):
Submit

Add sales_product Here:
Sale ID 4000: Thu Jan 11 2018 00:00:00 GMT-0800 (PST) **this is a dynamically populated list of sales from the database.**

Product ID: 3000: Fish Food - 1 lb **this is a dynamically populated list of products from the database.**

Quantity
Submit

© Tropical Fish Emporium

Change Log Page

CRUD Step & Purpose: N/A, track updates/changes overtime

| Home | Customers | Employees | Sales | Products | Manager | Change Log |

Tropical Fish Emporium

Step 5 to Step 6 Draft

These changes were made based on feedback from peer review during step 5 and updates to the site to include UPDATE and DELETE functionalities.

- UPDATE and DELETE:
  - UPDATEs to the employee table are now implemented. When viewing employees from the "Employee" page, an individual employee may be selected by clicking the Update button.
    - A new page is rendered, a GET is performed to populate the form fields
    - These forms are editable and changes can be made to the database by clicking on the "Update Employee" button.
  - DELETE is now functional on the `customers` page.
    - Clicking the delete button next to a customer will remove it from the database.
    - The corresponding foreign key in sales for customer ID is 'nulled' when this happens
- The Manager page now has two dynamically populated drop down lists.
  - Under the Sales section on the Manager page, all employees and customers in the database now appear in drop down lists, making it easy to add a sale and maintain valid foreign keys.
- A modification to the DDL file was made to ALTER the sales table to allow its customerID field to become NULL if the corresponding custmer is deleted from the database.
- For the sake of simplification, we have decided to only allow DELETEs from the `customers` table and UPDATEs from the `employees` table.

Step 4 to Step 5 Draft

These changes were made based on feedback from peer review during step 4 and updates to the site to include a backend that handles INSERTs and SELECTS.

- A DELETE SQL command was added for the products_sales intersection table.
- We recognize that the addition of a sales_product row(s) to the database should only occur after a sale is entered.
  - The saleID of the sale will be the saleID of the entries to sales_products.
  - The number of entries to sales_products for any given sale will be equal to the number of unique products that are part of that order.
- Upon loading the Customers, Employees, Sales or Products pages, the database is queried and all results found in that table are displayed.
- On the Manager page, entries may be made to each of the database's tables.
  - You can check that these results were added by going back to corresponding page. Results are ordered by ID in descending order, so your entry should be at the top of the table.

Step 3 Final to Step 4 Draft

These changes were made based on feedback from grading and team collaboration to refine the HTML template and best prepare our database outline for implementation.

- firstName and lastName attributes were added to the Employee table
- A constraint that the combined firstName and lastName of an Employee or Customer must be unique.
  - Although it is possible in the real world to have two people with the same first and last name, we felt this design decision would make the database more interesting by providing a UNIQUE key.
- The customerId and employeeId foreign key attributes of the Sales table were shortened to eID and cID.
- The saleId and productId foreign key attributes of the Sales_Products table were shortened to sID and pID.
- Added "See All" option to each table page. This will show every entity within that table.
- Adding a sale in the Manager tab will now also ask the user how many products were associated with that sale and will generate that number of entry boxes so that the sales_products intersection table gets populated. This would normally be automated by the system, but is manual for simplicity and demonstration purposes.

Step 3 Draft to Final Changes

These changes were made for the Final Edition (Step 3) based on peer and TA feedback:

Peer Feedback & Changes Made Log

Step 6: Implement UPDATE and DELETE operations

6.1 Changes Made

- Based on the peer review feedback, a code review and cleanup was done to better organized things. Time permitting, we may break the index.js file into smaller files.
- We have decided not to use bootstrap though as we do not feel that our website needs a cosmetic overhaul. Additional formatting and cleanup to the handlebars files (html) has been done to provide better continuity.

6.2 Peer Feedback

Tom Barabasz
1. Your group is on the right track. It looks like you've put some time into CSS and styling. To make it easier, I suggest using a CSS framework like Bootrstrap. It makes life a whole lot easier.
2. I also found the site UI a bit confising. I was not sure what and how to update. I'd suggest seperating some of the functionalities. As a user, I dislike scrolling all the way to the bottom just to find the submit button. Think of users as very lazy beings and try to make it easy for them to use the site.
3. Take a look at this code https://github.com/knightsamar/cs340_sample_nodejs_app It's a good example on how to structure the server side code. My group used it. Having a good structure in place makes it easier to stay organized. Good luck

Step 5: Implement CREATE + READ operations

5.1 Changes Made

- We added a delete to the sales_products table
- We changed variable names to prevent syntactical errors in js

5.2 Peer Feedback

Greg Noetzel

Bonus points for including the overview on the front page.

1. Are the queries syntactically correct?
   a. Yep! All queries currently written are syntactically correct.
2. Are there queries providing all functionalities as required by the CS340 Project Guide ? What query is missing ? What needs to be fixed?

       a. You are missing a delete for the sales_products table.
3. Do the queries cover the relationships as required by the CS340 Project Guide?
       a. All relationships are accounted for.

DDL:

4. Is the SQL file syntactically correct?
       a. I was able to import without the error you indicated by un-checking the "Enable foreign key checks" under "other options" in the import tab on myPHPadmin. Hope that helps.
5. Are the data types appropriate considering the description of the attribute in the database outline?
       a. Yep! I noticed that your product IDs are not auto-incrementing but all other IDs are. I wasn't sure if that was intentional so I wanted to mention it.
6. Are the foreign keys correctly defined when compared to the Schema?
       a. Yep! All foreign keys look properly defined.
7. Are relationship tables present when compared to the ERD/Schema?
       a. Yep! Intermediary table is intermediating.

Ruben Sanduleac

1. Are the queries syntactically correct?
       a. Yes, based on the files provided the queries are all syntactically correct.
2. Are there queries providing all functionalities as required by the CS340 Project Guide ? What query is missing? What needs to be fixed?
       a. Yes, currently there are all select and insert queries for each of the tables that are needed. There is also the required update query. There is one delete query missing for sales_product table.
3. Do the queries cover the relationships as required by the CS340 Project Guide?
       a. Yes, based on the information given the queries cover the relationship requirements.

DDL:

4. Is the SQL file syntactically correct?
       a. Yes, the SQL file is syntactically correct. I had one issue importing the file into phpMyAdmin. I had error #1451

```
#1451 - Cannot delete or update a parent row: a foreign
key constraint fails

-- EMPLOYEES ~~

DROP TABLE IF EXISTS employees
```

5. Are the data types appropriate considering the description of the attribute in the database outline?

      a. The database outline was not found in the zip file. The website did provide a clear representation of how it supposed to look like. The data types seem to be correct based on each attribute that is present.

6. Are the foreign keys correctly defined when compared to the Schema?
   a. Like the database outline, the schema was also not included in the submission. Based on the SQL files it looks like the foreign keys are correctly set up in the present files. I'm not sure why, but I had an issue when I imported the file. I think the foreign key is not configured correctly for Employees table. Can't know for sure because the schema is missing.

7. Are relationship tables present when compared to the ERD/Schema?
   a. The ERD and schema were not present in the submission. Based on the SQL website and tables that were created it does look like the relationships were set up correctly. Just I had an issue with the foreign keys in Employee table.

Website:

The website is built very well. The layout is set up in very logic manner. Thus, making it very user-friendly. I had no issues in navigating between different pages and testing the different tables. and easy to navigate. Overall, everything looks excellent.

Adam Slusser

1. Is the SQL file syntactically correct?
   a. Yes it is with the exception of the INSERT INTO sales_products VALUES which gave me the error "Cannot add or update a child row: a foreign key constraint fails"

2. Are the data types appropriate considering the description of the attribute in the database outline?
   a. The outline wasn't included in the zip file but based on browsing their website everything appears to be correct.

3. Are the foreign keys correctly defined when compared to the Schema?
   a. The Schema wasn't included in the files submitted. But judging from the website and the tables in myphp everything appears to be correct.

4. Are relationship tables present when compared to the ERD/Schema?
   a. I'm not able to compare the ERD to the Schema as these weren't submitted with their files.

DML:

5. Is the SQL file syntactically correct?
   a. Yes everything appears to be working correctly.

6. Are the queries providing all functionalities as required by the CS340 Project Guide?
   a. All the requirements appear to have been met from the queries that the group provided.

7. What query is missing? What needs to be fixed?
   a. There don't appear to be any missing or that need to be fixed.

8. Do the queries cover the relationships as required by the CS340 Project Guide?
   a. All the queries do cover the required relationships.

Shile Song

1. Is the SQL file syntactically correct?
   a. I can't import SQL file in the phpmyadmin. I look around the code and I find the error appear in that not set the `saleID` in class `sales` as a key.
2. Are the data types appropriate considering the description of the attribute in the database outline?
   a. It looks all tables follow the relationship base on the browsing.
3. Are the foreign keys correctly defined when compared to the Schema?
   a. Yes, every foreign key correct define follow the schema.
4. Are relationship tables present when compared to the ERD/Schema?
   a. Yes, these relationship tables present good with the ERD.

DMQ:

5. Are the queries syntactically correct?
   a. The DDL can't run in the server, so I don't test the dmq.
6. Are there queries providing all functionalities as required by the CS340 Project Guide ? What query is missing ? What needs to be fixed?
   a. The queries providing all follow the guild. They got select, insert, delete, update.
7. Do the queries cover the relationships as required by the CS340 Project Guide?
   a. Yes, they follow the guide.

Step 3: Design HTML Interface

3.1 Changes Made

- Create dynamic drop down menu (Piazza @170).
  - When ADDing a new Sale, have the option to select a customer from a drop down list. This list will be populated by using a SELECT to query all customers and populate the drop down.
  - We were previously a bit confused as to what would constitute a dynamically populated list, but the TA's forum post on Piazza clarified this.
- Show FK attributes of Sales_Products any time a Sale or Product is queried and make sure the INSERT (add) page for these two tables updates the Sales_Products intersection table as well.
  - During our first implementation we were not sure how to show foreign keys being added to our intersection table, but solved this by adding some example data to the Sales and Products tabs of the website.

- ○ Now, when clicking submit in either of these tabs, in addition to sample data for a Sale or Product being shown, a new table is shown called "Sales_Products" showing entries (foreign keys) related to the queried Sale or Product.
- Additionally, we received feedback from our TA on Project Step 2 noting that we had productId as an attribute of a Sale in both the Schema and ERD. While these had previously been removed from the outline, we had forgotten to remove them from the Schema and ERD. This has been remedied.
  - ○ We also noticed that SaleId was an attribute of the Customers entity on the ERD and Schema, which was also an error. This attribute has been removed from these images.

3.2 Other Updates

- Added productId as a search element for products.
  - ○ It seems that it will be useful to search (query) for a product by its ID number in addition to its name and other attributes.
- Example search for a product changed to just return a single product. As if the user had searched for "Clown Fish" or productId "2".
  - ○ To simulate searching for a single product, the example results were modified.
- Example search for a sale changed to return two sales generated by employee 14 (as if the user had searched for Employee 14's sales on a certain date.
  - ○ The corresponding Sales_Products table now shows all entries related to saleId 1 and 23.
    - ■ saleId #1 contains three different entries for the three products that were sold as part of that transaction.
    - ■ saleId #23 contains 1 entry for the single product that was sold during that transaction.

3.3 Peer Feedback

Casey Levy

1. Does the UI utilize a SELECT for every table in the schema? In other words, data from each table in the schema should be displayed on the UI. Note: it is generally not acceptable for just a single query to join all tables and display them.
   a. Yes it seems as if it does based on the current site and documentation. Each page has its own table and information matching what is given in the project's Schema.
2. Does at least one SELECT utilize a search/filter with a dynamically populated list of properties?
   a. There are search options on all pages except for the Manager page.
3. Does the UI implement an INSERT for every table in the schema? In other words, there should be UI input fields that correspond to each table and attribute in that table.

a. Yes, nearly every table so far has an INSERT. There are multiple tables to add or update employee data, as well as product data, among other details.

4. Does each INSERT also add the corresponding FK attributes, including at least one M:M relationship? In other words if there is a M:M relationship between Orders and Products, INSERTing a new Order (e.g. orderID, customerID, date, total), should also INSERT row(s) in the intersection table, e.g. OrderDetails (orderID, productID, qty, price and line_total).

   a. I personally don't see it within the site, but this is achieved based on the project's documentation.

5. Is there at least one DELETE and does at least one DELETE remove things from a M:M relationship? In other words, if an order is deleted from the Orders table, it should also delete the corresponding rows from the OrderDetails table, BUT it should not delete any Products or Customers.

   a. I didn't see a DELETE on the current version of the site, but based on the documentation, direction of the site, and the goal of the database, I can assume one would be implemented in the future.

6. Is there at least one UPDATE for any one entity? In other words, in the case of Products, can productName, listPrice, qtyOnHand, e.g. be updated for a single ProductID record?

   a. Yes, there is. The site offers a function to update employee details. One small addition could possibly be to be able to update product details, such as if the price has changed or the manufacturer has changed the product's name.

7. Is at least one relationship NULLable? In other words, there should be at least one optional relationship, e.g. having an Employee might be optional for any Order. Thus it should be feasible to edit an Order and change the value of Employee to be empty.

   a. It seems that in their documentation that there is at least one NULLable relationship or an attempt to implement one. i.e. There can't be a Sale without a Customer.

8. Do you have any other suggestions for the team to help with their HTML UI?

   a. Everything looks really good! The only suggestion I have is perhaps spacing some of the tables/forms a little further from the others on pages that have multiples, for easier readability. But other than that, the site looks great so far.

Yuxiao Huang

1. Does the UI utilize a SELECT for every table in the schema?

   a. Yes, each table has its own corresponding table for showing the information. And each web page has an unique table based on your documentation. You don't need to do any change for the content of this part.

2. Does at least one SELECT utilize a search/filter with a dynamically populated list of properties?

   a. Yes, I've seen many search boxes under your web pages except your Manger page.But you might not need to do that under Manger page.

3. Does the UI implement an INSERT for every table in the schema?

   a. Yes, each table has an INSERT button. Customer, employee, Product and so on tables need INSERT button. And you did that very clear.

4. Does each INSERT also add the corresponding FK attributes, including at least one M:M relationship?
   a. I did not find out that in your web pages. I get them in your documentations, so I think you will finish this part in the future work.
5. Is there at least one DELETE and does at least one DELETE remove things from a M:M relationship?
   a. I'm not sure, because I did not see any DELETE under any page of your website. But I think you will add them around INSERT buttons.
6. Is there at least one UPDATE for any one entity?
   a. Yes, in the "Employees" and the "Sales" pages, the UPDATEd table will be shown after the user types their information and clicks "Submit".
7. Is at least one relationship NULLable?
   a. I found that in your documentation, the relationship NULLable is to do something. For example, there cannot be a Sale without a Product.
8. Do you have any other suggestions for the team to help with their HTML UI?
   a. If you wanna create a whole website, you are supposed to do some colorful stuff. Like, set a background picture or change the font. In general, your website looks very clear and the new user can utilize it very quickly.

Will Dang

1. Does the UI utilize a SELECT for every table in the schema? In other words, data from each table in the schema should be displayed on the UI.
   a. Yes, the UI does utilize a SELECT for every table in the schema, and does so in accordance to the project documentation provided.
2. Does at least one SELECT utilize a search/filter with a dynamically populated list of properties?
   a. Yes, multiple pages utilize a search function with dynamically populated list of properties, except for the manager page (which doesn't require this function).
3. Does the UI implement an INSERT for every table in the schema? In other words, there should be UI input fields that correspond to each table and attribute in that table.
   a. Every table in the schema has an INSERT button for necessary attributes such as customers, products, etc.
4. Does each INSERT also add the corresponding FK attributes, including at least one M:M relationship? In other words if there is a M:M relationship between Orders and Products, INSERTing a new Order (e.g. orderID, customerID, date, total), should also INSERT row(s) in the intersection table, e.g. OrderDetails (orderID, productID, qty, price and line_total).
   a. I did not notice this when checking it, but according to your documentation, you plan to implement this so I am assuming that will be done sometime in the future.
5. Is there at least one DELETE and does at least one DELETE remove things from a M:M relationship? In other words, if an order is deleted from the Orders table, it should also delete the corresponding rows from the OrderDetails table, BUT it should not delete any Products or Customers.

      a. The DELETE only becomes available after having submitted the data for display. This function was seen from the "Customers," "Employees," "Sales," and "Products" pages.

6. Is there at least one UPDATE for any one entity? In other words, in the case of Products, can productName, listPrice, qtyOnHand, e.g. be updated for a single ProductID record?
   a. Yes, there is an UPDATE function, particularly in the pages for Employees and Sales. After the user inputs information and clicks submit, the updated table should return.

7. Is at least one relationship NULLable? In other words, there should be at least one optional relationship, e.g. having an Employee might be optional for any Order. Thus it should be feasible to edit an Order and change the value of Employee to be empty.
   a. There is at least one NULLable relationship according to the provided draft. For instance, the relationship between Sales and Customers (cannot have sale without a customer).

8. Do you have any other suggestions for the team to help with their HTML UI?
   a. The only suggestion I have is to figure out how to add corresponding fk attributes for the INSERT. Otherwise, I think that your website looks easy to read and very clear. Maybe add more stylistic component to make it more interesting visually (add a picture or two, etc).

Step 2: ERD & Schema

2.1 Changes Made

- We have removed the productId as an attribute of Sales as a result of this TA feedback.
  - We had intended to remove this, and the outline of the database reflects this, but our Schema and ERD had not been updated to reflect this change.
- Remove Stores as an entity.
  - We decided to step away from the idea of tracking sales, products, employees, and customers of different stores. Instead we are aiming to track the entire company as a whole, storeIDs only specified in employee and sales attributes.
  - This feedback was given to us by our TA during initial grading the project proposal and did not come from peer review. At first, we thought we'd want to retain this entity, but after seeing that it caused some confusion during peer review and after our team completed module 4, we think it is best to remove the Stores entity entirely.
- Remove Customer to Employee relationship.
  - Upon further consideration, an employee to customer relationship may not need to be explicit. This information could be found as both an employee and customer will be related to every Sale (by their respective ID numbers).
  - This removes a M:M relationship with one remaining between Products and Sales.

- Added clarification to outline descriptions
  - As a team, we have gone back through the outline and added more descriptions to the attributes and their constraints that needed it most.
- Mandatory participation of Product to Sales relationship and of Customer to Sales.
  - A change has been made so that a sale now must have 1 or many products instead of 0 or many.
  - We thought about it and decided to allow a customer to be in the system and not be associated with any Sales. For example, a customer may have set up a profile, but decided not to buy anything.

2.2 Other Updates

- We decided to remove the assignedTask attribute from the Employee entity as it would likely end up being a list of individual tasks. It would be difficult to conform to Normal Form 1 with this.
  - If future expansion of the database is needed, we will be able to add some relationship between a schedule and an employee, but this will only occur if time or a future assignment allows for it.
- In order to conform to NF 3, we reworked and simplified attributes.
  - The "workDays" of an employee violates normal form 1, so we removed this attribute.
  - We modified the startTime and stopTime attributes to not be dependent on information from the workDays field.
- Added a transactionDate attribute to the Sale entity so that the client can query the database to look at sales on a particular day.
- An intersection table called "Sales_Products" was added to relate the many-to-many relationship between Sales and Products. This means that Sales now has a 1:M relationship with Sales_Products and Products now has a 1:M relationship with Sales_Products.

2.3 Peer Feedback

Kyeong-nam Kim
1. Does the overview describe what problem is to be solved by a website with DB back end?
   a. Yes, the overview clearly describes the problem and solution to be implemented by a DB back end.
2. Does the overview list specific facts?
   a. Yes, the overview provides the numbers of specific data in a form of table.
3. Are at least four entities described and does each one represent a single idea to be stored as a list?
   a. Yes, there are 5 entities and each entity represent a single idea.
4. Does the outline of entity details describe the purpose of each, list attribute datatypes and constraints and describe relationships between entities?
   a. Yes, every entity has a description of its purpose, attributes along with datatypes and constraints, and relationships with different entities.
5. Are 1:M relationships correctly formulated? Is there at least one M:M relationship?

   a. Yes, 1:M relationships are correctly formulated. Two M:M relationships are well designed.
6. Is there consistency in a) naming between overview and entity/attributes b) entities plural, attributes singular c) use of capitalization for naming?
   a. Other than attribute hours which is singular in Stores, naming is consistent.

Daniel Yopp
1. Does the overview describe what problem is to be solved by a website with DB back end?
   a. Yes! Great idea and writeup! Its clear writer was enthusiastic about idea.
2. Does the overview list specific facts
   a. Yes. Description lists current store traffic as well as the workload the system will be expected to preform.
3. Are at least four entities described and does each one represent a single idea to be stored as a list?
   a. Yes. There are five total entities. Even though employees and customers are both people, their roles are so different that I believe they can be stored as different entities.
4. Does the outline of entity details describe the purpose of each, list attribute datatypes and constraints and describe relationships between entities?
   a. Yes. Each entity is thoroughly documented.
5. Are 1:M relationships correctly formulated? Is there at least one M:M relationship?
   a. I think the relationships are appropriate. The customer to employee may be a bit of a stretch but I think its ok.
6. Is there consistency in a) naming between overview and entity/attributes b) entities plural, attributes singular c) use of capitalization for naming?
   a. Yes, naming is consistent. Entities are capitalized and plural. Attributes are camelCase and singular.

Christopher Brown
1. Does the overview describe what problem is to be solved by a website with DB back end?
   a. Yes, the database acts as a data aggregator for a chain of fish stores. It tracks the relationships between the 3 locations and it's employees and customers.
2. Does the overview list specific facts
   a. Yes, the overview gives specifics on the number of stores, employees, transactions, and customers.
3. Are at least four entities described and does each one represent a single idea to be stored as a list?
   a. Yes, there are 5 different entities to be tracked by the database. They are the Stores, the Employees, Customers, Sales, and Products.
4. Does the outline of entity details describe the purpose of each, list attribute datatypes and constraints and describe relationships between entities?
   a. The outline gives a list of attributes and some constraints for each entity, but each could use a brief description of its purpose.
5. Are 1:M relationships correctly formulated? Is there at least one M:M relationship?

      a.   Yes, there are 1:M relationships for Customers and Employees to Sales, and Stores to Employees. There are several M:M relationships, although I am a little confused of the purpose of the Customer:Employee relationship repeated outside of Sales.

6.   Is there consistency in a) naming between overview and entity/attributes b) entities plural, attributes singular c) use of capitalization for naming?

      a.   Yes, the schema, outline, and ERD are consistent in naming and capitalization. However, on the ERD, I think the participation is mandatory for at least the Product:Sales relationship (a SaleID must have at least 1 ProductID), and likely Stores:Employees (a store needs at least 1 employee) and Customer:Sales as well (are they really a customer if you've never sold them anything?)

2.4 TA Feedback

- ProductID should not be a field in Sales; the Sales_Product tables helps detail a Sale as-is.